

Two-dimensional global manifolds of vector fields

Bernd Krauskopf^{a)}

Department of Engineering Mathematics, University of Bristol, Bristol BS8 1TR, United Kingdom

Hinke Osinga^{b)}

Control & Dynamical Systems, Caltech 107-81, Pasadena, California 91125

(Received 16 November 1998; accepted for publication 18 May 1999)

We describe an efficient algorithm for computing two-dimensional stable and unstable manifolds of three-dimensional vector fields. Larger and larger pieces of a manifold are grown until a sufficiently long piece is obtained. This allows one to study manifolds geometrically and obtain important features of dynamical behavior. For illustration, we compute the stable manifold of the origin spiralling into the Lorenz attractor, and an unstable manifold in ζ^3 -model converging to an attracting limit cycle. © 1999 American Institute of Physics. [S1054-1500(99)02403-9]

The computation of stable and unstable manifolds for vector fields is of direct practical use in applications, because these manifolds form boundaries of basins of attraction and their intersection gives rise to chaotic dynamics. So far only one-dimensional manifolds of vector fields are readily computable, which limits the study to two-dimensional vector fields. However, many interesting systems, for example the well-known Lorenz and Chua systems, are three-dimensional vector fields. In this paper we are concerned with the technically challenging case of computing two-dimensional stable and unstable manifolds of three-dimensional vector fields. Combined with visualization techniques, this constitutes a new powerful tool for a complete exploration of the dynamical behavior of three-dimensional physical models. Our method starts near a saddle point or periodic orbit and grows the manifold as a sequence of closed curves until a prespecified arclength is reached. We take special care of maintaining a prescribed mesh quality, which is achieved by solving suitable boundary value problems.

I. INTRODUCTION

Vector fields are the mathematical models of choice in numerous applications. Of particular importance are vector fields with a three-dimensional phase space, because they are the lowest dimensional continuous time models that can exhibit extremely complicated behavior and chaos. Well-known examples from applications are Lorenz's simple weather model,^{1,2} Chua's model of a chaotic electronic circuit,³ and periodically forced oscillators; see, e.g., Refs. 2 and 4 and further references therein. The basic task is to understand the dynamics of such a vector field. First one needs to find all equilibria and periodic orbits, which can be attractors, repellers, or of saddle type. The challenge is then to find the stable and unstable manifolds of the equilibria and periodic orbits of saddle type, because they organize the glo-

bal dynamics of the system. For example, the stable manifolds form the boundaries between different attractors, and an intersection between stable and unstable manifolds usually leads to complicated dynamics and chaos. Global bifurcations and routes to chaos can be identified by tracing how stable and unstable manifolds change as parameters vary. This makes it possible to gain geometric insight into the underlying dynamics of the system under consideration.

It is generally not possible to find stable and unstable manifolds other than by numerical computation. One-dimensional stable and unstable manifolds (of saddle points) consist of two distinct orbits that can be computed by integrating two suitable initial conditions. However, a higher-dimensional stable or unstable manifold consists of infinitely many orbits, and a finite collection of such orbits typically does not give an acceptable picture of the manifold. More sophisticated methods are needed, and we refer to Sec. II for an overview of the literature.

In this paper we present a new method for computing two-dimensional stable and unstable manifolds of equilibria and periodic orbits in a three-dimensional state space. This is illustrated with manifolds of the Lorenz system and the ζ^3 -model in Sec. IV; see also EPAPS (Ref. 22) for color figures and animations showing how these manifolds are grown. We stress that the numerical ideas presented in this paper can be formulated so that they allow for computing a general k -dimensional manifold of an n -dimensional vector field. Here we only consider two-dimensional manifolds of three-dimensional vector fields for the following reasons. First, as mentioned above, three-dimensional vector fields are an important class. In particular, we plan to use our method to study transitions to chaos in the three-dimensional rate equation model of a semiconductor laser with optical injection.^{5,6} Second, it is natural to develop, implement, and explain a new algorithm in a simple setting. A theoretical treatment of the general case, including a proof of correctness, can be found in Ref. 7. Important issues of data management still need to be addressed in order to implement even the case of three-dimensional manifolds, and this remains work for the future. Finally, by working in a three-dimensional state space we avoid the substantial difficulty of

^{a)}Electronic mail: b.krauskopf@bristol.ac.uk

^{b)}Electronic mail: hinke@cds.caltech.edu

visualizing a complicated object in a high-dimensional space.

We now introduce some mathematical notation. Our object of study is a three-dimensional vector field

$$\dot{x} = f(x), \quad (1)$$

where $x \in \mathbb{R}^3$ and $f: \mathbb{R}^3 \mapsto \mathbb{R}^3$ is sufficiently smooth. The dynamics is determined in this case by the stable and unstable manifolds of equilibria and periodic orbits of saddle type. Recall that the stable and the unstable manifold of a saddle point x_0 are defined as

$$W^s(x_0) = \{x \in \mathbb{R}^3 \mid \lim_{t \rightarrow \infty} \phi^t(x) = x_0\},$$

$$W^u(x_0) = \{x \in \mathbb{R}^3 \mid \lim_{t \rightarrow -\infty} \phi^t(x) = x_0\},$$

where ϕ^t is the flow of Eq. (1). Note that the stable manifold $W^s(x_0)$ is the unstable manifold of x_0 when time is reversed. The stable and unstable manifolds of a periodic orbit Γ of saddle type are defined in the same way. To keep the exposition simple we explain our algorithm for a saddle point x_0 ; the case of a saddle periodic orbit requires only a minor modification as is explained in Sec. III E. The stable and unstable manifold theorem⁸ states that in a neighborhood of x_0 there exist local stable and unstable manifolds $W_{\text{loc}}^s(x_0)$ and $W_{\text{loc}}^u(x_0)$, which are tangent to the stable and unstable eigenspaces $E^s(x_0)$ and $E^u(x_0)$ of the Jacobian $Df(x_0)$ of Eq. (1), respectively. Either the stable or the unstable manifold of any saddle point x_0 is two-dimensional. For convenience we discuss the case of a two-dimensional unstable manifold $W^u(x_0)$ in the sequel.

Our algorithm grows $W^u(x_0)$ in circles starting with an initial circle on $E^u(x_0)$. Its important feature is that it computes the two-dimensional unstable manifold as a mesh in such a way that the quality of the mesh is guaranteed. For each mesh point r on the last circle we do the following. We define a two-dimensional plane through r perpendicular to the last circle, and determine a new point on $W^u(x_0)$ in this plane at a suitable distance from r . Finding this new point amounts to solving a boundary value problem, which is done by a shooting approach. The algorithm works when x_0 has eigenvalues with largely differing modulus, as well as in the case of complex conjugate eigenvalues. As a particular feature we detect convergence to an attractor, which allows us to compute the entire unstable manifold in this case.

The organization of the paper is as follows. We start with an overview of other methods in Sec. II. The algorithm is explained in Sec. III, and its performance is illustrated with two examples in Sec. IV.

II. OTHER METHODS

There are three different ideas to compute a two-dimensional unstable manifold of a vector field. In some sense the most straightforward idea, which is also our approach, is to start with a small circle of points on $E^u(x_0)$ around x_0 and grow the manifold as a one-parameter family of (topological) circles. However, simply integrating a circle has the disadvantage that this circle typically does not grow

uniformly in all radial directions. Since one can only evolve a finite number of points on any given circle one quickly encounters the problem that points drift apart, leading to a mesh of poor quality.

Johnson *et al.*⁹ tackle this problem by integrating each of a finite number of nicely distributed points on a circle until a prescribed arclength is reached. This defines a new circle of points. In a second step extra points are added if necessary by using interpolation, after which the points are redistributed equally spaced along the piecewise linear circle to again obtain a nice distribution of points on the next circle. The procedure is repeated until a sufficient piece of the manifold has been obtained. Since the dynamics on the manifold controls the distance between the points on the new circle before the redistribution of mesh points, not much can be said about the accuracy of the interpolated points.

In the methods of Guckenheimer and Worfolk^{10,11} divergence between points in the integration is avoided by rescaling the vector field. In this way it can be achieved that the tangential component is practically zero at each point along a circle. This changes the dynamics, but not $W^u(x_0)$ as a set. All these points are then integrated until a prescribed distance is reached to obtain the next circle. Again, a sufficient piece of the manifold can be grown by repeating this procedure. Their method requires that the vector field always has a radial component pointing outward with respect to the last circle.

Doedel¹² computes two-dimensional manifolds with a very different method using continuation. To start, he computes an orbit of a prescribed arclength (or arclength times integration time) by integrating as the initial condition a point on a ray through x_0 on $E^u(x_0)$ very close to x_0 . This orbit is then continued with the boundary value solver of AUTO,¹³ where the angle of the ray is the continuation parameter. In this way, the orbit sweeps over $W^u(x_0)$, which gives very accurate results. A possible disadvantage is that this method produces an abundance of mesh point near x_0 .

Finally, there is the approach of considering the time t map of the flow of Eq. (1) for some fixed $t > 0$, and use an algorithm designed for maps. This is done, e.g., in Ref. 14. Another interesting method for discrete systems is the one by Dellnitz and Hohmann,^{15,16} who cover the manifold with three-dimensional boxes. They start with a covering of $W_{\text{loc}}^u(x_0)$ with sufficiently small boxes that they obtain with a subdivision method. In a second step the covering of $W_{\text{loc}}^u(x_0)$ is extended to obtain a covering of a first finite piece of $W^u(x_0)$ in a particular window of interest. A two-dimensional approximation of $W^u(x_0)$ can be obtained by connecting the center points of the boxes in the covering with triangles. The accuracy of the computation is directly related to the diameter of the boxes, so that a large number of boxes may be required. We remark that, in general, using the time t map instead of the vector field itself leads to longer computation times.

III. THE ALGORITHM

In this section, which is necessarily quite technical in nature, we describe in detail how the algorithm works. The

reader may find it helpful to look ahead to the examples in Sec. IV. We explain all steps for the case of the unstable manifold $W^u(x_0)$ of a saddle point x_0 of the vector field Eq. (1). The stable manifold can be computed in the same way by considering Eq. (1) for reversed time. Only a minor change is needed for the case of the unstable manifold $W^u(\Gamma)$ of a periodic orbit of saddle-type Γ , which is explained in Sec. III E. We finally discuss numerical properties of the algorithm in Sec. III F.

A. The unstable manifold of a saddle point

In the present setting, one eigenvalue λ^s of x_0 is real and negative, and the other two eigenvalues λ_1^u and λ_2^u have positive real parts (and may be real or complex). We start with a linear approximation of the local unstable manifold $W_{\text{loc}}^u(x_0)$ and use the fact that the forward orbit of any point on $W_{\text{loc}}^u(x_0)$ lies on the global manifold $W^u(x_0)$. It is known⁸ that $W^u(x_0)$ is tangent at x_0 to the plane $E^u(x_0)$ spanned by the (generalized) eigenvectors of λ_1^u and λ_2^u . The starting data for our algorithm is a finite set of points on a small circle in $E^u(x_0)$ of radius δ about x_0 .

The global manifold $W^u(x_0)$ is generated by computing more and more discrete circles, each of which consists of a finite number of mesh points. The manifold $W^u(x_0)$ is then represented as the triangulation given by these mesh points. Hence, adding a discrete circle corresponds to adding a band of triangles to the computed piece of $W^u(x_0)$.

Let C_r denote the last computed discrete circle of mesh points, and let \widehat{C}_r denote the continuous circle obtained by linearly connecting the mesh points in C_r . The key step is to add a new discrete circle, denoted by C_b , at a particular distance Δ from C_r . The problem is to ensure that the mesh points of C_b are nicely distributed. To achieve this, we define a linear foliation $\{\mathcal{F}_r\}_{r \in C_r}$, that is, a set of (half) planes \mathcal{F}_r , transversal to the last circle \widehat{C}_r . We now consider the one-dimensional intersection curves of $W^u(x_0)$ with $\{\mathcal{F}_r\}_{r \in C_r}$. The choice of $\{\mathcal{F}_r\}_{r \in C_r}$ has to be such that locally the intersection of $W^u(x_0)$ with each leaf \mathcal{F}_r is a curve of sufficient arclength; compare Ref. 14. In each leaf \mathcal{F}_r we find a point in $W^u(x_0)$ at distance Δ from the base point r by solving a boundary value problem. This set of points forms the new discrete circle C_b , which indeed lies at distance Δ from C_r with nicely distributed mesh points. More details of this procedure are given below in Sec. II B; see also Fig. 1.

If mesh points come too close to each other, or move apart too much we update the mesh, which is discussed in Sec. III C. The algorithm also detects and deals with convergence to an attractor, and this is explained in Sec. III D.

B. Adding a new circle

As the first step we need to find a foliation $\{\mathcal{F}_r\}_{r \in C_r}$ such that $W^u(x_0)$ intersects each leaf \mathcal{F}_r , $r \in C_r$, in a curve locally near \widehat{C}_r . A plane in three-dimensional space is uniquely defined by a normal vector, and we define \mathcal{F}_r by its normal vector

$$(r_{\text{right}} - r) + (r - r_{\text{left}}) = r_{\text{right}} - r_{\text{left}},$$

Algorithm GLOBALIZE

Input: f , right-hand-side of vector field;
 x_0 , saddle point of f ;
 C_r, δ , finite mesh on circle $C(x_0, \delta)$ in $E^u(x_0)$;
 Δ , guess for distance between circles;
 L_{arc} , arclength to be computed;
Output: mesh on $W^u(x_0)$ of arclength approximately L_{arc} from x_0

Band array $[0, \dots, 3][|C_r|]$ of Point;
 Comment: representation for \widehat{C}_p , \widehat{C}_r , and \widehat{C}_b

Leaf array $[|C_r|]$ of Point;
Delta, ARC real;
Admissible boolean;

Begin
 1 Band[0] = x_0 ; Band[1] = C_r ;
 Comment: initially \widehat{C}_p is circle with 0 radius
 2 Print(Band[0], Band[1]);
 Comment: data is stored in file
 3 ARC = δ ;
 4 while ARC < L_{arc} do
 5 for $r \in C_r$ do
 6 Leaf[r] = $r_{\text{right}} - r_{\text{left}}$;
 Comment: Determine vector normal to \mathcal{F}_r
 7 Band[2][r] = BVP(Band[1], Leaf[r], r, Delta);
 8 Admissible = ACCURACY(Band);
 9 if Admissible then
 10 ARC = ARC + Delta;
 11 Band[0] = Band[1]; Band[1] = Band[2];
 12 Print(Band[0], Band[1]);
 13 adjust Delta;
End.

FIG. 1. The algorithm GLOBALIZE in pseudocode. The procedure BVP is described in Sec. III B, and the procedure ACCURACY is outlined in Sec. III C.

where r_{left} and r_{right} are the neighboring mesh points on the left and right of r , respectively. Note that this normal vector can be considered as tangent to \widehat{C}_r , at r . The next step is to find a point $b_r \in W^u(x_0) \cap \mathcal{F}_r$ at a prescribed distance Δ from r . The choice of Δ is influenced by the local curvature of $W^u(x_0)$. In fact, Δ is a guess and it is checked at a later stage whether this guess is in accordance with the required accuracy; see Sec. III C.

The key idea is that the next point $b_r \in \mathcal{F}_r$ can be found by continuing the following boundary value problem until the required distance Δ is reached. We require an orbit $\{\phi^t(q_r(\tau)) | t \in [0, \tau]\}$ satisfying the boundary conditions

$$\begin{aligned} \phi^0(q_r(\tau)) &\in \widehat{C}_r, \\ \phi^\tau(q_r(\tau)) &\in \mathcal{F}_r. \end{aligned} \quad (2)$$

For convenience we denote the initial condition on \widehat{C}_r by $q_r(\tau)$, and the final point in \mathcal{F}_r by $b_r(\tau) = \phi^\tau(q_r(\tau))$. The trivial solution $q_r(\tau) = r$ and $b_r(\tau) = r$ with $\tau = 0$ satisfies the boundary condition (2). We continue this solution in the continuation parameter τ while we monitor the distance

$$\|b_r(\tau) - r\|. \quad (3)$$

Note that this distance is zero for $\tau = 0$. The continuation stops when for some (first) τ_0 we have $\|b_r(\tau_0) - r\| = \Delta$. We set $b_r = b_r(\tau_0)$ and are done; see Fig. 2.

In practice, we accept the point $b_r(\tau_0)$ if $\|b_r(\tau_0) - r\| \in [(1 - \epsilon)\Delta, (1 + \epsilon)\Delta]$ for some prespecified ϵ . In the examples of Sec. IV we used $\epsilon = 0.01$. This continuation part of the algorithm is performed in the procedure BVP in Fig. 1. In the present implementation we use a shooting method as a boundary value solver.

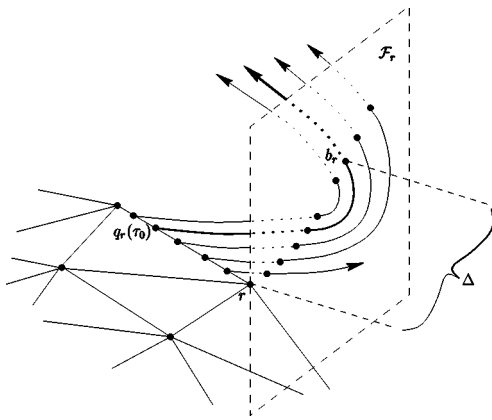


FIG. 2. The next point b_r in the plane \mathcal{F}_r through r is found by continuation in τ of orbits $\{\phi^t(q_r(\tau)) | t \in [0, \tau]\}$ starting on the last circle \widehat{C}_r , and ending in \mathcal{F}_r . The point $q_r(\tau_0)$ is uniquely defined as the point on the last circle for which $b_r = \phi^{\tau_0}(q_r(\tau_0))$ lies at distance Δ from r .

After the computation of C_b , we need to check whether the band of triangles between C_r and C_b accurately approximates $W^u(x_0)$. It is possible that \widehat{C}_b lies too far from \widehat{C}_r , or that we need more mesh points in C_b . This is described in the next section.

C. Mesh adaptation

Our algorithm automatically updates the mesh after each step. The following is taken into consideration to determine whether the mesh is still accurate. First we check whether the distance Δ from \widehat{C}_b to \widehat{C}_r is sufficiently small. It may be that \widehat{C}_b should be closer to \widehat{C}_r in order to get an accurate approximation of $W^u(x_0)$. Next, we check whether \widehat{C}_b satisfies the conditions for the mesh quality. If necessary, extra points are added, or removed. The mesh adaptation is done in the procedure ACCURACY in Fig. 1.

After performing the step in Sec. III B, both circles \widehat{C}_r and \widehat{C}_b consist of the same number of mesh points. We form lines from points in C_r to corresponding points in C_b . If these lines accurately follow the one-dimensional curves $W^u(x_0) \cap \{\mathcal{F}_r\}_{r \in C_r}$ locally near \widehat{C}_r , then we accept Δ as a good estimate. Otherwise, we reject C_b , decrease Δ , and repeat the step in Sec. III B. The accuracy of one-dimensional curves is controlled using the technique described in Ref. 17, which is based on controlling the angle between three successive points together with the product of this angle and the distance between the last two of the three points; compare also Ref. 18. Our next guess for Δ is based on these considerations as well. Since the angle is defined using three successive points, we need to keep track of the circle before \widehat{C}_r in the data structure; see also Fig. 1, where this circle is called C_p . Note that initially this circle has radius 0.

Once we have accepted \widehat{C}_b as a new circle at a proper distance from \widehat{C}_r , we check how well \widehat{C}_b approximates $W^u(x_0)$. The accuracy of \widehat{C}_b is controlled by the distance between neighboring mesh points. We prespecify minimal and maximal distances between two neighboring mesh

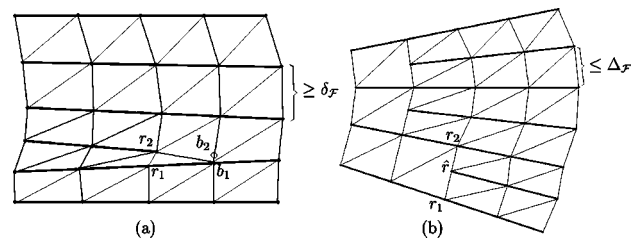


FIG. 3. Mesh adaptation by removing (a) and adding (b) leaves.

points, denoted by $\delta_{\mathcal{F}}$ and $\Delta_{\mathcal{F}}$, respectively. Let r_1 and r_2 be two neighboring mesh points in C_r , and let b_1 and b_2 denote the corresponding points in C_b .

It is most important that b_1 and b_2 are not too close to each other, since then, in the next step, the leaves through these points might cross at a distance less than Δ from b_1 or b_2 . When the mesh points are not properly ordered, the algorithm will be confused when it computes the next circle. Hence, if $\|b_2 - b_1\| < \delta_{\mathcal{F}}$, we wish to delete a mesh point. Therefore, we set $b_2 = b_1$, and drop b_2 as a mesh point in the next step; see also Fig. 3(a).

If $\|b_2 - b_1\| > \Delta_{\mathcal{F}}$, then the mesh is not accurate enough. We wish to add an extra point to C_b . For this, we first add the point

$$\hat{r} = \frac{1}{2}(r_2 + r_1)$$

to C_r . We next define $\mathcal{F}_{\hat{r}}$ and compute $b_{\hat{r}}$ by solving the corresponding boundary value problem; see Fig. 3(b).

D. Convergence to an attractor

In the event that $W^u(x_0)$ has finite arclength it may not be possible to find points at distance Δ . Moreover, the arclength generally differs from leaf to leaf. Therefore, when we detect convergence, we allow different Δ steps in different leaves. This means that we let Δ depend on the mesh points $r \in C_r$.

For a particular leaf \mathcal{F}_r we find $q_r(\tau)$ and $b_r(\tau) = \phi^{\tau}(q_r(\tau))$ during the continuation of the boundary value problem. Recall that we are monitoring the distance $\|b_r(\tau) - r\|$ until we find the point $b_r(\tau_0)$ for which this distance is Δ . If the manifold converges, we may have that $\|b_r(\tau) - r\|$ is always smaller than Δ , even for $\tau \rightarrow \infty$. If this is the case, we allow for different Δ steps in different leaves.

In practice, we put a maximum τ_{\max} on τ . If we find that $\|b_r(\tau) - r\| < \Delta$ for all $0 < \tau \leq \tau_{\max}$, we take the best candidate, that is, we set $b_r = b_r(\tau_{\max})$, such that

$$\|b_r - r\| = \Delta_r < \Delta,$$

and Δ_r is as large as possible. This way, we assure a relatively smooth decrease and increase of Δ_r along \widehat{C}_b . The algorithm detects convergence in a leaf \mathcal{F}_r if the best possible new point b_r lies at distance Δ_r smaller than a prescribed minimum. Subsequently, the algorithm stops computing new points in this leaf. If convergence is detected in all leaves, the algorithm stops altogether and reports convergence to an attractor.

E. The unstable manifold of a periodic orbit

Only a simple change in starting data is needed to compute the unstable manifold of a periodic orbit Γ of saddle type. At each point r on Γ there is a well-defined unstable linear direction $v^u(r)$. Both Γ and its unstable directions can be found by looking at suitable Poincaré sections. By choosing a mesh on Γ , we define an initial circle as the set of points at distance δ from the mesh points r on Γ in the direction of $v^u(r)$. Growing this initial circle is now done in exactly the same way as for a saddle point; compare Ref. 14.

F. Properties of the algorithm

Our method is independent of the dynamics on $W^u(x_0)$ without a need for rescaling the vector field. Also, we do not require that the vector field points outward with respect to the last circle, as is required in Refs. 10 and 11. When adding new mesh points, that is, new planes, interpolation is only used within the prescribed mesh distance, so that the interpolation error is always under control; compare Ref. 9. The mesh is adapted to the local curvature of $W^u(x_0)$, so that the number of mesh points is minimal without sacrificing the accuracy. The speed of the computation is largely determined by the time needed for integration, which depends on the dynamics. Solving boundary value problems is more time consuming than integrating from a fixed initial point, but we find that the algorithm is quite fast.

There are two main sources of errors in the computation. First, there is the initial error that one makes by picking starting data on $E^u(x_0)$ at a prescribed distance δ from x_0 . Second, there is the interpolation error between mesh points. The algorithm is correct in the sense that both these errors tend to zero when the accuracy of the computation is increased. For the rather technical proof we refer to Ref. 7, where also the general algorithm for k -dimensional stable and unstable manifolds in an n -dimensional phase space can be found.

IV. EXAMPLES

Our algorithm does not depend on the dynamics on the manifold. For example, it is not influenced by the ratio of the eigenvalues of the linear part at the saddle point. This is demonstrated in Sec. IV A, where two stable eigenvalues of the origin differ by a factor of order 10. Furthermore, the algorithm has no problems dealing with complex conjugated eigenvalues, which is illustrated in Sec. IV B. There, we also show how the algorithm handles convergence to an attracting limit cycle. All figures have been rendered with the package GEOMVIEW.¹⁹ Animations of growing manifolds can be found via the link in Ref. 22.

A. The Lorenz system

The well-known Lorenz system¹ is the vector field

$$\begin{aligned} \dot{x} &= \sigma(y-x), \\ \dot{y} &= \rho x - y - xz, \\ \dot{z} &= xy - \beta z. \end{aligned} \quad (4)$$

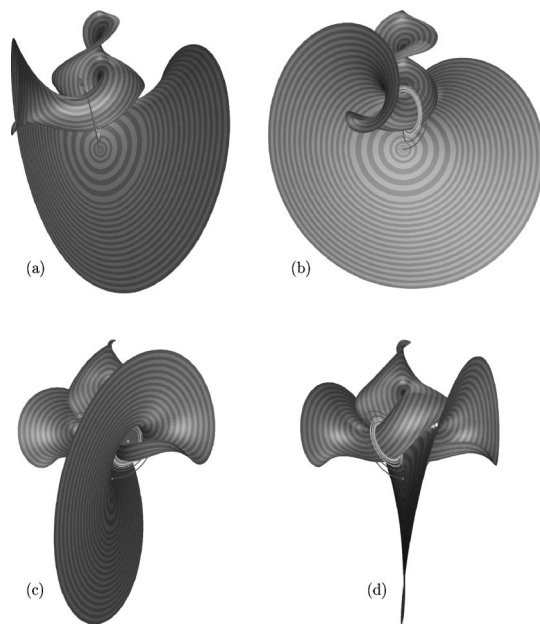


FIG. 4. The stable manifold $W^s(0)$ of the Lorenz system viewed along the y -axis (a), with rotations about the z -axis (vertical axis) of 40° (b), 120° (c), and 140° (d), respectively.

The origin is always an equilibrium, and for the standard parameters $\sigma=10$, $\rho=28$, and $\beta=2\frac{2}{3}$ it has a stable manifold for which the corresponding eigenvalues are $\lambda^s = -\beta$ and $\lambda^{ss} \approx -22.8$. This stable manifold is known to have a complicated structure, because it spirals into the butterfly-shaped strange attractor without actually intersecting it. Note that the z -axis is invariant and part of the stable manifold.

We compute $W^s(0)$ using the following accuracy parameters. The vector field is integrated with a fixed time step Runge–Kutta order four integrator, taking a maximum of 150 steps. The time step is 0.001 initially and is adapted as points are integrated further away from the equilibrium such that always at least two integration steps are taken to determine the next circle. The distance between circles is at least $\Delta_{\min}=0.01$, and it is controlled by the accuracy conditions explained in Ref. 17 ($0.3 < \alpha < 0.2$, and $0.1 < \Delta\alpha < 1.0$). We start with an initial circle of 20 points at distance $\delta=1.0$ from the origin. The distance between leaves is kept at a maximum of $\Delta_{\mathcal{F}}=2.0$, and is at least $\delta_{\mathcal{F}}=0.5$. The final circle contains 885 points. The result of the computation is shown in Figs. 4–6. For completeness, the one-dimensional unstable manifold $W^u(0)$ is shown as well (computed with

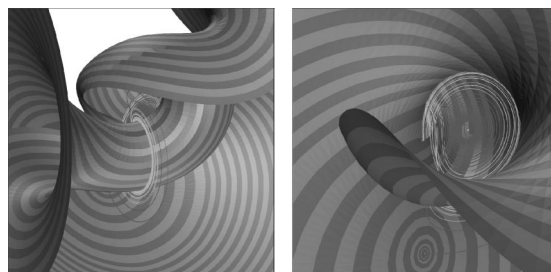


FIG. 5. Two close-up views showing how the stable manifold $W^s(0)$ spirals into the Lorenz attractor.

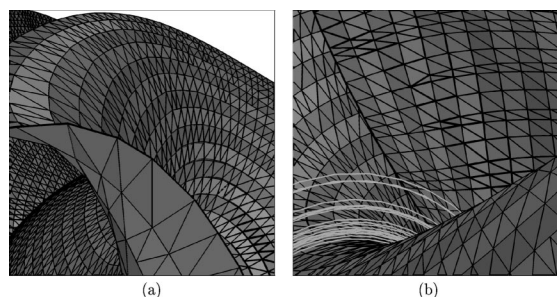


FIG. 6. Mesh points are added (a) and removed (b) in the course of the computation.

the package DSTOOL²⁰). Figure 4 shows four views of the entire manifold computed up to about arclength 100. The two close-up views of $W^s(0)$ in Fig. 5 show the complicated interaction of the manifold and the attractor. Obviously, many leaves are added during this computation, but also some leaves are removed as they come too close to neighboring leaves. Both adaptations are illustrated in Fig. 6.

B. ζ^3 -model

The following vector field is the ζ^3 -model²¹:

$$\begin{aligned}\dot{x} &= y, \\ \dot{y} &= z, \\ \dot{z} &= \alpha x - x^2 - \beta y - z,\end{aligned}\quad (5)$$

where $\alpha = 2.5$ and $\beta = 2$. This system has two equilibria, one at the origin and one at $(\alpha, 0, 0)$. For this choice of parameters, both equilibria are of saddle type. We concentrate on the equilibrium $(\alpha, 0, 0)$ with eigenvalues $\lambda^s \approx -1.15$ and $\lambda^u \approx 0.075 \pm 1.47i$ and consider its two-dimensional unstable manifold, which converges to an attracting limit cycle.

As in Sec. IV A, we use a fixed time step order four Runge–Kutta integrator, with a maximum of 1000 steps. Due to the spiraling behavior on the manifold, the next circle will typically be much more than 1 integration step away, so we take the time step 0.01 and keep it fixed during the entire computation. Further accuracy parameters are $\Delta_{\mathcal{F}} = 0.1$, $\delta_{\mathcal{F}} = 0.025$, $\Delta_{\min} = 0.01$, $0.2 < \alpha < 0.3$, and $0.001 < \Delta\alpha < 0.1$. We start with 40 points on an initial circle of radius $\delta = 0.1$. The algorithm first grows circles uniformly until the manifold gets close to the attracting limit cycle; see Fig. 7(a). As there is slow radially outward growth near the attracting limit cycle, it becomes harder to find points at distance Δ from the boundary. When this happens, the algorithm allows for a variable growth rate per leaf, such that the convergence process can be completed; see Fig. 7(b). The resulting mesh is shown in Fig. 7(c). We assume convergence in a leaf when the Δ step in this leaf cannot be taken larger than 10^{-3} . Note that the algorithm is not affected by the spiraling behavior of orbits on the manifold. Even if the vector field points radially inward at certain points on the boundary of the manifold, the next circle can be found properly; see Fig. 7(d).

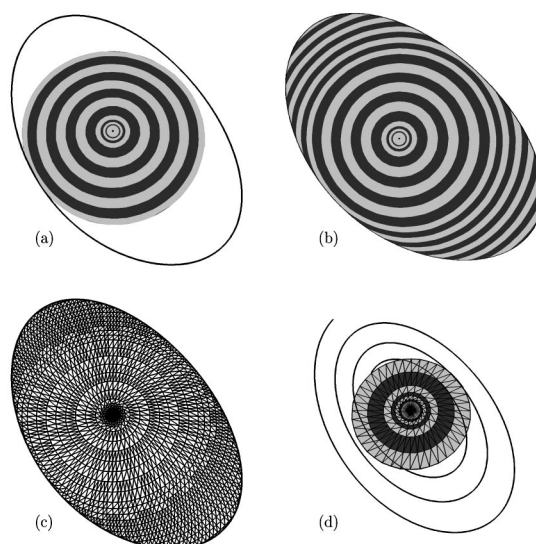


FIG. 7. The unstable manifold $W^u(\alpha, 0, 0)$ of the ζ^3 -model grows with the same speed in each leaf until it hits an attracting limit cycle (a). Then the convergence process starts until the entire manifold has been computed (b), resulting in the mesh shown in (c). The vector field does not always point outward with respect to the last circle as is illustrated by the orbit of a mesh point in (d).

ACKNOWLEDGMENTS

This work was initiated during the authors' participation in the Special Year on Emerging Applications of Dynamical Systems 1997/98 at the Institute for Mathematics and its Applications, Minneapolis. We thank W.-J. Beyn, M. Dellnitz, E. J. Doedel, J. Guckenheimer, and M. E. Johnson for helpful discussions, and the IMA for its hospitality and support. B.K. thanks the California Institute of Technology and H.O. the University of Bristol for its hospitality. H. O. was supported by AFOSR/DDRE MURI AFS-5X-F496209610471. Additionally, material can be found in the EPAPS.²²

¹E. N. Lorenz, "Deterministic nonperiodic flows," *J. Atmos. Sci.* **20**, 130–141 (1963).

²S. H. Strogatz, *Nonlinear Dynamics and Chaos* (Addison–Wesley, Reading, MA, 1994).

³R. N. Madan, *Chua's Circuit: a Paradigm for Chaos*, World Scientific Series on Nonlinear Science, Series B, Vol 1 (World Scientific, Singapore, 1993).

⁴J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields* (Springer, New York, 1983).

⁵B. Krauskopf, N. Tollenaar, and D. Lenstra, "Tori and their bifurcations in an optically injected semiconductor laser," *Opt. Commun.* **156**, 158–169 (1998).

⁶G. H. M. van Tartwijk and D. Lenstra, "Semiconductor lasers with optical injection and feedback," *Quantum Semiclassical Opt.* **7**, 87–143 (1995).

⁷B. Krauskopf and H. M. Osinga, "Global manifolds of vector fields: The general case," *IMA J. Num. Analysis* (submitted).

⁸J. Palis and W. de Melo, *Geometric Theory of Dynamical Systems* (Springer, New York, 1982).

⁹M. E. Johnson, M. S. Jolly, and I. G. Kevrekidis, "Two-dimensional invariant manifolds and global bifurcations: Some approximation and visualization studies," *Numer. Algorithms* **14**, 125–140 (1997).

¹⁰J. Guckenheimer and P. Wofolk, "Dynamical systems: Some computational problems," in *Bifurcations and Periodic Orbits of Vector Fields*, edited by D. Schlomiuk (Kluwer, Dordrecht, 1993), pp. 241–277.

¹¹P. Wofolk (private communications).

¹²E. J. Doedel (private communications).

¹³E. J. Doedel, A. R. Champneys, T. R. Fairgrieve, Yu. A. Kuznetsov, B. Sandstede, and X. J. Wang, "AUTO97 continuation and bifurcation soft-

- ware for ordinary differential equations," available via <ftp://ftp.cs.concordia.ca/pub/doedel/auto/> (1997).
- ¹⁴B. Krauskopf and H. M. Osinga, "Globalizing two-dimensional unstable manifolds of maps," *Int. J. Bifurcation Chaos Appl. Sci. Eng.* **8**, 483–503 (1998).
 - ¹⁵M. Dellnitz and A. Hohmann, "A subdivision algorithm for the computation of unstable manifolds and global attractors," *Numer. Math.* **75**, 293–317 (1997).
 - ¹⁶M. Dellnitz, A. Hohmann, O. Junge, and M. Rumpf, "Exploring invariant sets and invariant measures," *Chaos* **7**, 221–228 (1997).
 - ¹⁷B. Krauskopf and H. M. Osinga, "Growing 1D and quasi 2D unstable manifolds of maps," *J. Comput. Phys.* **146**(1), 404–419 (1998).
 - ¹⁸D. Hobson, "An efficient method for computing invariant manifolds of planar maps," *J. Comput. Phys.* **104**, 14–22 (1993).
 - ¹⁹M. Phillips, S. Levy, and T. Munzner, "GEOMVIEW: An Interactive Geometry Viewer," *Notices Am. Math. Soc.* **40**, 985–988 (1993). This software and the accompanying manual are available at <http://www.geom.umn.edu/>.
 - ²⁰A. Back, J. Guckenheimer, M. R. Myers, F. J. Wicklin, and P. A. Worfolk, "DSTOOL: Computer assisted exploration of dynamical systems," *Notices Am. Math. Soc.* **39**, 303–309 (1992).
 - ²¹A. Arneodo, P. Couillet, E. Spiegel, and C. Tresser, "Asymptotic chaos," *Physica D* **14**, 327–347 (1985).
 - ²²See AIP Document No. EPAPS: ECHAOEH-9-024903 for animations and color figures for the Lorenz system and the ζ^3 -model. EPAPS document files may be retrieved free of charge from AIP's FTP server (<http://www.aip.org/pubservs/paps.html>) or from <ftp.aip.org> in the directory /epaps/. For further information, e-mail: paps@aip.org or fax: 516-576-2223.